

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1 (Currently Amended): In a computer system under control of an operating system comprising modules of code and an operating system kernel, a dynamically loadable stub module, associated with a dynamically loadable kernel module (DLKM) for dynamically loading modules into the kernel whereby access to the operating system is provided, comprising:

a base stub module;

means for defining DLKM data structures and wrapper functions wherein the means for defining the DLKM data structures and wrapper functions comprises an autoload statement;

means for defining metadata structures;

means for allowing dynamic loading by DLKM infrastructures; and

means for generating a dynamically loadable stub module object file.

Claim 2 (Original): The dynamically loadable stub module of claim 1, wherein the means for defining the DLKM data structures and wrapper functions comprises:

struct mod_type_data;

struct modlink;

struct modwrapper; and

strut mod_operations.

Claim 3 (Cancelled):

Claim 4 (Currently Amended): The dynamically loadable stub module of claim 3 ~~1~~, wherein the means for defining load and unload routines comprises statements class, and one of stub funcname retfunc, unstub funcname retfunc argnword, and wstub funcname retfunc.

Claim 5 (Original): The dynamically loadable stub module of claim 1, wherein the means for defining load and unload routine comprises:

<module_name>_stub_load(); and

<module_name>_stub_load().

Claim 6 (Original): The dynamically loadable stub module of claim 1, wherein the means for defining metadata structures comprises module version, type, definition, states, and loadtime.

Claim 7 (Original): The dynamically loadable stub module of claim 1, wherein the means for defining metadata structures comprises a developer-supplied modmeta file.

Claim 8 (Original): The dynamically loadable stub module of claim 7, wherein the modmeta file is compiled by a modmeta compiler to produce a stub_modmeta.c file.

Claim 9 (Original): The dynamically loadable stub module of claim 7, wherein metadata is supplied from the associated DLKM.

Claim 10 (Original): The dynamically loadable stub module of claim 1, wherein the dynamically loadable stub module is included in a kernel data space.

Claim 11 (Original): The dynamically loadable stub module of claim 1, wherein the dynamically loadable stub module is capable of being statically linked to a kernel executable.

Claim 12 (Original): The dynamically loadable stub module of claim 1, wherein the data structures comprise:

struct mod_stub_modinfo; and
struct mod_stubinfo.

Claim 13 (Original): The dynamically loadable stub module of claim 12, further comprising the stub routines that use the data structures to manipulate stack frames to transfer control from the dynamically loadable stub module to the associated DLKM.

Claim 14 (Original): The dynamically loadable stub module of claim 1, wherein the means for allowing dynamic loading by DLKM infrastructures comprises an ELF section.

Claim 15 (Original): The dynamically loadable stub module of claim 1, wherein the associated DLKM is a miscellaneous module.

Claim 16 (Currently Amended): In a computer system under control of an operating system comprising modules of code and an operating system kernel, a method for dynamic loading of a stub module to provide operating system access comprising:

defining DLKM data structures and wrapper functions for the stub module comprising providing an autoloader statement comprising class, and one of stub funcname retfunc, unstub funcname retfunc argword, and wstub funcname retfunc;

defining load and unload routines for the stub module;

defining metadata structures for the stub module;

providing for dynamic loading of the stub module by DLKM infrastructures; and

generating a dynamically loadable stub module object file.

Claim 17 (Original): The method of claim 16, wherein defining DLKM data structure and wrapper functions comprises defining:

struct mod_type_data;

struct modlink;

struct modwrapper; and

strut mod_operations.

Claim 18 (Cancelled).

Claim 19 (Original): The method of claim 16, wherein defining metadata structures comprises providing a developer-supplied modmeta file.

Claim 20 (Original): The dynamically loadable stub module of claim 19, further comprising compiling the modmeta file to produce a stub_modmeta.c file.

Claim 21 (Currently Amended): A computer-readable medium for use in a computer system under control of an operating system comprising modules of code and an operating system kernel, the computer-readable medium having computer code to implement autoloader stub modules, the code when executed comprising the following steps:

Defining DLKM data structures and wrapper functions for the stub module using an autoloader statement;

Defining load and unload routines for the stub module;

Defining metadata structures for the stub module;

Providing for dynamic loading of the stub module by DLKM infrastructures; and

Generating a dynamically loadable stub module object file.